



**Page manquante
(colophon)**

abc

INFORMATIQUE

**VOLUME
8**

**EDITIONS
ATLAS**

Éditions Atlen, Bruxelles
Finabuch s.a., Éditions Transalpines, Mezzovico
Éditions Atlas Canada Ltée, Montréal Nord

Sommaire



Le marché

- Contrôler l'ordinateur directement par la pensée semble *a priori* relever de la science-fiction. Et pourtant, c'est non seulement faisable, mais aussi une réalité 1681
- Le domaine de l'intelligence artificielle représente sans doute l'aspect le plus passionnant de l'informatique. Voici des techniques de base qui peuvent s'appliquer à votre micro 1701
- A l'aide de l'énigme classique du « rat dans le labyrinthe », voyons quelques principes fondamentaux des techniques d'intelligence artificielle permettant de résoudre des problèmes 1721
- Nous poursuivons notre étude sur l'intelligence artificielle par l'élaboration de programmes susceptibles de prévoir le déroulement d'une partie d'échecs et de décider des mouvements 1748
- Le secteur de la recherche de la S.N.C.F. s'intéresse depuis plusieurs années au traitement de la parole, et réalise actuellement des expériences de synthèse et de reconnaissance vocales. 1758
- Tout en examinant de près la planification stratégique en intelligence artificielle, nous allons étudier certaines améliorations ainsi que d'autres stratégies requises par les jeux de hasard 1761
- Elle ne donne pas dans le spectacle et, pourtant, elle ne compte pas moins de deux cent soixante clubs en France. Il s'agit de l'association Microtel, qui met l'informatique à la portée de tous 1766
- Les systèmes experts sont des programmes hautement structurés susceptibles de tenir en grande partie le rôle de conseil et de diagnostic pour des professionnels 1781
- Comment mettre au point un système capable d'apprendre par lui-même? Passons en revue les critères nécessaires à la création d'un tel système, et examinons Beagle, un « moteur d'inférences » 1801
- Dans cet article consacré aux machines capables d'apprendre, nous proposons un programme intitulé Gene, qui permet de trouver de meilleures solutions à des problèmes de réseaux assez complexes 1828
- Il existe des systèmes capables d'entendre et de parler; mais il se passera encore beaucoup de temps avant qu'ils ne puissent converser « naturellement » 1841
- « Voir » et « comprendre », deux aspects des choses difficiles à envisager séparément. Mais c'est nécessaire pour qu'un ordinateur « comprenne » ce qu'il sera amené à « voir » 1861
- Continuons notre étude des systèmes informatiques de vision en voyant comment il est possible d'identifier des caractéristiques inscrites à l'intérieur d'un ensemble de formes 1881
- La mise au point effective de robots intelligents se heurte encore à de très nombreux problèmes. Voyons d'un peu plus près le fossé entre la théorie et la pratique 1901



Mots de passe

- Nous continuons la création d'une base de données. Nous passons de l'utilisation des fonctions à l'intérieur des paramètres au développement d'un algorithme informel 1684
- Nous allons voir des concepts en PASCAL, tels la déallocation, les pointeurs, les sauts et les structures liées. Ces derniers nous permettent de traiter des fichiers de taille indéterminée 1714
- Pour en finir ici avec le PASCAL, nous allons nous attarder sur divers aspects de ce langage que nous n'avons pas encore vus, en particulier sur la description des données 1726
- PROLOG est un langage à la fois très utile et convivial. Nous allons, au cours d'une série d'articles sur PROLOG, définir sa logique de base et retracer son histoire 1752
- PROLOG résout les problèmes en ayant recours à une base de données de structures appelées « faits » et en appliquant ces derniers aux propositions par essais successifs. Étudions donc la nature déclarative du langage 1772
- Le déroulement d'un programme PROLOG ne suit pas le même schéma que celui des langages comme le BASIC ou le PASCAL. Arrêtons-nous sur la procédure arborescente de recherche de PROLOG 1792
- Nous allons analyser comment PROLOG incorpore des effets annexes et des propriétés « non logiques » tels que les opérations « couper » et « échec » 1806
- PROLOG possède la propriété de se servir de ses propres programmes comme données. Cela le destine tout particulièrement à être utilisé en programmation pour l'intelligence artificielle 1832
- Fondé sur des listes pouvant représenter des données ou des fonctions, le LISP, très répandu dans l'intelligence artificielle, permet d'adapter le langage à pratiquement toutes les applications 1856
- Nous montrons ici comment créer des listes à l'intérieur des listes et mettons en évidence l'importance des instructions conditionnelles et des fonctions utilisateur en langage LISP 1864
- Voyons pourquoi il peut être utile de savoir comment le LISP représente des structures de données et illustrons par quelques exemples diverses fonctions ... 1895



- Les lacunes au niveau du stockage à accès rapide ont limité le succès de l'Amstrad. L'unité de disquette Amstrad DDI-1 lui permet d'accroître son potentiel à peu de frais 1689
- L'un des premiers constructeurs de micros, la Tandy Corp., n'a pas véritablement réussi à s'imposer sur le marché. Le compatible Tandy 1000 marquera-t-il un tournant? 1729
- Les interfaces du Spectrum ne permettent pas d'utiliser la plupart des imprimantes. Voici néanmoins quatre solutions avantageuses par rapport à la ZX 1741
- L'interface Bananarama, produite par Castle Associates, destinée principalement au marché de l'éducation, est un périphérique peut-être un peu limité pour une utilisation sur micro domestique 1770
- Avec son nouveau micro 130XE, Atari à la fois assure la compatibilité avec ses prédécesseurs XL et conserve ses excellentes fonctions graphiques et sonores. Une machine très compétitive 1789
- Les commerçants viennent de plus en plus nombreux à l'informatique. La S.E.I.T.A. propose aux débits de tabac des services et des moyens adaptés à leurs besoins 1808
- Le progiciel Echo Music, destiné au Commodore 64, tente de pallier l'absence de produits accédant directement à la puce sonore des micros. Mais ce logiciel laisse un peu à désirer 1821
- Avec l'introduction du Quick Data Drive 8500 de Phonemark, qui utilise un nouveau support, les « wafers », le problème du stockage sur le C64 est réellement amélioré 1852
- Même si l'Apricot F1e est d'abord une machine de gestion, il peut très bien réussir sur le marché de l'éducation et auprès des petites entreprises 1869
- Véritable 16 bits, doté du processeur 80186 travaillant à 8 MHz, le Goupil G4 associe le MS-DOS au GWBASIC, et il est entièrement compatible aux niveaux logiciel et matériel 1873
- Comparons le Cray-1, l'ordinateur le plus puissant du monde, et un micro construit autour d'un Z80. Une application servira de référence. Les différences sont considérables 1884
- Depuis l'arrivée du Macintosh sur le marché, et de sa souris, cette dernière fait des petits. La SMC Supplies Magic Mouse est la première souris destinée au C64 1890
- L'Omni-reader, d'Oberon International, est l'un des derniers OCR à microprocesseur pouvant lire un texte imprimé et le charger dans un micro possédant une interface RS232C 1909
- Dix-huit mois après avoir lancé son premier micro-ordinateur familial, Philips, s'étant rallié au standard MSX avec les VG8010 et 8020, devenait ainsi le cheval de Troie des Japonais 1918



- Nous allons intégrer les quatre événements majeurs qui restent à programmer : attaque des pirates, tempête, gouvernail rompu, apparition d'une île.... 1692
- Nous avons déjà vu que de nombreux événements modifient le cours du voyage. Mais l'équipage peut aussi se mutiner 1712
- Votre navire vient d'aborder le Nouveau Monde. Il est possible maintenant de commercer avec les indigènes, pourvu que vous n'entriez pas en guerre avec eux 1724
- Si vous n'avez commis aucune erreur grave, vous voilà dans le Nouveau Monde. Ayez un peu le sens des affaires pour revenir confortablement en Europe ... 1746
- Votre voyage dans le Nouveau Monde touche à sa fin. Avant de partir, vous devez décider de participer ou non aux troubles politiques locaux, ce qui pourra vous valoir d'énormes profits (ou de tout perdre...) 1774
- Les derniers articles consacrés à notre jeu de simulation traiteront de variantes BASIC qui lui permettront de tourner sur Amstrad et Spectrum 1794
- Nous mettons un terme à notre jeu de simulation consacré à un voyage dans le Nouveau Monde en donnant ici la seconde partie du listage complet 1834
- Avec le jeu d'échecs, le go, venu d'Extrême-Orient, a fasciné les chercheurs en intelligence artificielle. Nous allons mettre au point un programme joueur de go 1846
- Voici les listages qui permettront d'adapter le programme de go au C64, au Spectrum et à l'Amstrad 464/664. Ils sont consacrés à l'affichage du plateau de jeu et comportent plusieurs modules 1886
- Définissons les modules qui nous permettront de jouer au go. Nous ferons usage d'une structure « en pile » des données, ainsi que du concept de récursivité 1905



Jeux

- Le voleur, représenté par un masque noir, s'est échappé en emportant le magot. Ce jeu, écrit par Pierre Monsaut, utilise un thème bien connu dans l'informatique ludique 1695
- Qui ne connaît pas ce jeu ? Voilà une bonne occasion pour les possesseurs d'un micro C64 de le traduire en BASIC et de remplacer la feuille par l'écran 1708
- Le micro EXL 100 d'Exelvision n'est pas oublié dans notre rubrique « Jeux ». Des programmes écrits par Pierre Monsaut ont été adaptés à cet ordinateur. En voici un 1732
- Pour les possesseurs du micro-ordinateur EXL 100, voici le programme d'un jeu écrit par Pierre Monsaut. Dans le style « des chiffres et des bombes » 1769
- Voici un jeu d'action, écrit par Pierre Monsaut, qui est toujours d'actualité. Mais n'oubliez pas que ce programme est destiné au micro-ordinateur EXL 100 1788
- Ce jeu d'action existe pour de nombreux modèles de micro-ordinateurs. Nous vous présentons ici le programme destiné au MO5 de Thomson 1820
- Rien de bien difficile dans ce programme écrit par Pierre Monsaut pour le Commodore 64. Mais prudence, n'oubliez pas de l'enregistrer avant de le faire tourner 1840
- Vous êtes le commandant d'un vaisseau de guerre et vous arrivez dans une galaxie inconnue... C'est le début d'une grande aventure sur votre ordinateur Atari 1845
- Votre micro-ordinateur vous invite à une partie de chasse au canard. Laissez-vous tenter par ce jeu écrit par Pierre Monsaut pour le Commodore 64 1872
- Pour que ce programme puisse tourner sur votre micro-ordinateur, il est nécessaire que ce dernier soit conforme à la norme MSX (son, couleur, graphisme) 1889



Boîte à outils

- Voici un programme destiné à commander les mouvements du robot à partir du Spectrum, *via* l'interface construite précédemment comme nous l'avons fait avec le Commodore 64 1696
- Nous allons modifier le logiciel mis au point pour notre bras-robot. Nous verrons également des mises au point pour améliorer la régularité des mouvements du bras 1706
- Nous commençons la conception et la construction d'une interface MIDI pour permettre au C64 de communiquer avec plusieurs instruments électroniques numériques 1815
- Construisons pour le C64 une interface lui permettant de communiquer avec la gamme croissante d'équipements compatibles MIDI 1823
- Seconde étape de construction de l'interface MIDI : connexion des broches de la puce ACIA aux lignes de données, de commande et d'alimentation de l'ordinateur 1848
- Après avoir construit l'interface MIDI, nous attirons l'attention sur un autre aspect tout aussi important du logiciel qui nous permettra de jouer de la musique « en temps réel » 1866
- Avec l'interface MIDI, concevons un programme permettant à l'ordinateur d'agir comme unité d'enregistrement numérique et comme unité d'exécution 1892



Logiciel

- Avant de faire l'acquisition d'un gestionnaire de base de données — qui n'est pas donné — il vous faut prendre en compte un grand nombre d'éléments techniques 1704
- Une équipe de linguistes de l'université de Paris VII a développé, en collaboration avec des informaticiens, un système de saisie et de traitement de texte en chinois 1709
- En combinant certaines caractéristiques des échecs et des graphiques à action rapide, Archon, d'Ariolasoft, met à l'épreuve l'habileté du joueur 1740
- Cet article est le premier d'une série qui étudiera en détail la création, la structure et la mise en œuvre d'un des systèmes d'exploitation les plus connus 1744
- Poursuivant notre étude du système CP/M, nous allons maintenant examiner les commandes « transitoires », qui se révèlent fort utiles lors de la gestion des fichiers et des périphériques 1764
- Voici un nouveau jeu produit par Euston Films, dans lequel vous jouez le rôle du fumeur de cigares Arthur, toujours à la recherche d'une bonne affaire et d'un profit rapide 1780

Nous poursuivons notre étude de CP/M en nous intéressant aux fichiers sur lesquels les commandes travaillent, et aux façons de les manipuler.....	1784	Nous présentons ici un aperçu de quatre logiciels générateurs de programmes et procédons à une étude détaillée du fonctionnement de deux d'entre eux : Sycero et The Last One	1854
Le scénario de ce jeu est sans doute familier, mais la commande du curseur par sélection d'icônes ajoute une dimension intéressante au jeu Shadowfire de Beyond Software	1800	Nous étudions les procédures des générateurs de programmes Sycero et The Last One. En insistant sur la préorganisation, ces systèmes vous aident à générer des programmes BASIC concis	1876
La commande SYSGEN permet à CP/M de se copier lui-même d'un disque à l'autre. Suivons le flux des informations à travers le système	1804	Le marché britannique étant avant tout tourné vers les possesseurs de cassettes, les programmeurs se sont vu distancer par les Américains, dont les logiciels sur disquettes sont plus complets	1880
Dans ce dernier article sur le système d'exploitation CP/M, nous étudions comment l'organisation interne de la mémoire de l'ordinateur est conçue afin d'optimiser la place disponible	1826	Vous connaissez le langage pédagogique LOGO. Voyons DR LOGO pour ordinateurs Amstrad et comparons-le avec l'original destiné à l'IBM PC ...	1912

Langage machine



Voici un programme pour le C64, qui permet d'afficher simultanément des graphiques haute résolution sur un écran fractionné — technique fréquemment employée dans les jeux d'aventures	1698	Attardons-nous sur les flux de données que le Spectrum envoie par ses canaux d'entrée/sortie vers l'imprimante et l'écran, ou reçoit à partir du clavier	1812
La puce dédiée au son du Commodore 64, ou puce SID, est virtuellement un synthétiseur de son. Avec un logiciel sophistiqué, elle peut même produire de la parole comme un robot	1717	Nous étudions les routines de ROM utilisées par le système d'exploitation du Spectrum pour accéder au système de fichier sur bande	1837
Nous explorons quelques-unes des routines en virgule flottante d'interpréteur BASIC pour le C64, en développant un programme graphique sur écran haute résolution	1736	L'UC Z80 offre au programmeur en langage machine trois modes d'interruption masquable, ce qui vous permet d'« emprunter » du temps au processeur en cas de besoin	1858
Cet article sur les graphiques 3-D (trois dimensions) sur Commodore 64 complète la conversion du listage BASIC et introduit des routines interpréteur en virgule flottante	1754	L'Interface 1 apporte au Spectrum Sinclair l'accès à quelques routines en langage machine qui peuvent nous servir dans nos programmes. Voici quelques-unes de ces routines	1878
Nous allons commencer à explorer les aspects du système d'exploitation du Sinclair Spectrum, en inspectant l'agencement de la table d'implantation en mémoire de l'ordinateur	1776	Après avoir vu les codes crochets utilisés par l'Interface 1, considérons les fonctions concernant l'exploitation des Microdrive, un périphérique bien utile du Spectrum	1898
Nous poursuivons notre série sur le SE du Spectrum par l'étude des canaux à travers lesquels le micro envoie des données à l'écran et à l'imprimante ZX ..	1797	L'Interface 1 fournit toutes sortes de routines utiles en langage machine Spectrum. Nous allons voir les codes crochets dont elle se sert pour contrôler le port série RS232	1914

Livre



Nous avons déjà présenté des livres permettant de mieux exploiter les possibilités des micros Thomson. Nous donnons ici une liste de livres pour le VG 5000 de Philips et l'Alice de Matra	1733	Pour aller plus loin en LOGO et découvrir les multiples facettes de ce langage particulièrement intéressant et formateur pour l'esprit, voici quelques livres	1818
Vous êtes déjà un fervent adepte de PASCAL, langage structuré et efficace; ou bien vous n'avez pas encore eu le courage ou l'occasion de vous lancer à sa conquête... Alors, voici des livres pour vous	1786	Voici une sélection de livres pour vous initier à ces étranges et merveilleuses machines que sont les micro-ordinateurs. Ils vous feront entrer dans le cercle des amateurs éclairés	1916

Errata

Volume 7

P. 1447, listage de la colonne 1, compléter les lignes :
5106 IF GF<> 0 THEN...
5145 IF VBS<> "TUER" THEN...
5175 IF F<>1 THEN...
5248 IF F<>3 THEN...
Colonne 2, encadré « Variantes de basic », ligne 23, remplacer « \$ » (au début de la ligne) par **TS**.
P. 1448, compléter ainsi les lignes du listage Digitaya :
3090 IF VBS<> "DONNER" THEN...
3160 IF F<>5 THEN...
3480 ...PEU A PEU...
3520 ...IF NS<>3 THEN...
3580 IF VBS<> "UTILISER" THEN...
5290 IF VBS<> "DONNER" AND VBS<> "OFFRIR" THEN...
P. 1456 et 1457, corriger ainsi le listage BBC Micro :
130 ... AND 192)<>neither_bumpers...
470 IF WX>1279 THEN...
490 IF WY>1023 THEN...
600 pd_ratio...:pa_ratio...
620 both_bumpers...:neither_bumpers...
610 right_bumper...:left_bumper...
700 ...pd_ratio...
760 ...pa_ratio...
P. 1477, encadré gris, ligne 9, lire : ...UNTIL AS="O"... (Lettre O.)
P. 1478-79, compléter les lignes du listage Digitaya comme suit :
2060 IF VAL(IVS(1,2))<>P THEN...
2068 ...THEN ZS="UN"
2090 IF F=0 THEN SNS=SNS+"...RIEN DU TOUT"
2515 ...THEN VBS=ISS...
2560 IF AS<>" " THEN 2590
3050 ...OR VBS="DENOMBRE"...
3530 IF DR5<>"N" AND DR5<>"E" AND DR5<>"S" AND DR5<>"O" THEN...
3570 IF DR5="O"...
3600 PRINT NNS:?"QU'EST-CE DONC?"
3650 IF MIDS(NNS...
3655 ...MIDS(NNS...
4330 GC=GC+1:IF GC>4 THEN...
4550 SNS=SNS+"ET SE RUE VERS VOUS"
4665 VBS="JE NE COMPRENDS PAS":...
5140 IF VF=1 THEN 5145:...
5175 IF F<>1 THEN SNS="ÇA NE SERVIRA A RIEN":...
5248 IF F<>3 THEN SNS="ÇA NE SERVIRA A RIEN":...
P. 1484, colonne 1, listage en bas de page, corriger la première et la dernière ligne :
PROGRAM Premier (résultat);
End.
P. 1485, colonne 2, ligne 40, remplacer « parenthèses » par accolades; ligne 43, lire : **Program Deux (input, output);** ligne 46 : **Message = 'Tapez un nombre';** ligne 48 : **nombre : Integer;** ligne 55 : **End.** (avec point final).
Page 1497, encadré gris, listage BBC Micro :
3890...:REM TOUCHE
P. 1498-1499, corriger le listage comme suit :
1720 IF ISS="REGARDE" THEN...
2110 PRINT NNS+"QU'EST-CE DONC?"
2170 REM**OBJET POSSEDE**
2320 ...:GOSUB 5880
2400 REM**OBJET POSSEDE?*2460 ... POSITION
2630 ...:IF AS<>"O" AND AS<>"N"...
2660 :
2700 IF F>7 THEN...
3480 ...MAGNETIQUE"
4030 IF HF=0 THEN...
4620 SNS=SNS+" 'ET', 'OU' ET 'NON'...
P. 1504, colonne 1, ligne 2, remplacer « Entier » par **Integer** et « Réel » par **Real**; ligne 3, remplacer « Booléen » par **Boolean**. Colonne 2, lignes 43 et 44, mettre « antérieur » et « postérieur » entre accolades ({}).
P. 1505, colonne 1, ligne 4, remplacer « vrai ou faux » par **true** ou **false**; lignes 7, 11 et 22, remplacer « faux » par **false**; ligne 8, remplacer « vrai » par **true**; ligne 19, lire **0.15** (au lieu de « 0,15 »); ligne 20, ajouter un deux-points après « 40 »; ligne 21, remplacer « » par ' '; ligne 31, lire : **genre : real**; ligne 32, lire : **nombre : integer**; ligne 34, lire : **effectué : boolean**; ligne 50, lire un point-virgule au lieu du deux-points.
Colonne 2, ligne 2, remplacer « entrées, sorties » par **input, output**; remplacer partout les guillemets (") par des apostrophes (') dans les listages; ligne 8, remplacer « réel » par **real**; lignes 17 et 18, lire :
WriteLn ('est:',surface:10:3)
END.
Ligne 31, lire : **séparés par deux points** (au lieu de « une virgule »); ligne 63 : **WriteLn('Total':20, poids:1, 'tonnes').**
P. 1524, colonne 2, listage : 25 ... **COM-**

MENCE":GOTO 45.

P. 1526, colonne 1, ligne 19, écrire « ne rien faire » entre accolades ({}).
Colonne 2, ligne 30, ajouter **bleu** entre « vert » et « jaune »; lignes 35-36, remplacer « valeur entière » par **Integer**.
P. 1527, colonne 1, ligne 7, lire : **lettre:char**; ajouter un point-virgule à la fin des lignes 57, 58, 59 et à la fin de la ligne 2 de la colonne 2; ligne 3, lire **END; [CASE]**; supprimer la virgule à la fin de la ligne 14.
P. 1528, colonne 1, ajouter un point-virgule après « END » (lignes 16 et 34).
Encadré jaune, corriger le listage comme suit :
PROGRAM Date (input, output); ligne 13, lire : **Mai**; ligne 14 : **Dec**; ligne 16 : **NomDuMois**; ajouter entre les lignes 16 et 17 : **Restent**; ligne 19 : **année:integer**; lignes 20 et 21 : **Symbole:char**; ligne 22 : **AnnéeBissextilé:boolean**; ligne 31 : **IF (mois>0) AND (mois<= 12)**; ligne 46 : **END [CASE]**; ligne 55 : **Mai,Jul,Aou,Oct,Dec:Restent...**; ligne 58 : **IF AnnéeBissextilé**; ligne 63 : **END; [CASE]**; ligne 65 : **WriteLn('Il y a'; Restent:1, 'dans');**; ligne 79 : **END; [CASE]**.
P. 1546, colonne 2, ligne 2, lire :
20 ...:"KILO":US(2)...
P. 1547, colonne 2, corriger le listage comme suit :
2086 ...:US(T);
2100 ...:US(T);...
2242 IF PS<>"O" AND...
2290 ...:"S DE "...
P. 1554, colonne 1, ligne 36, remplacer « réel » par **Integer** et « entier » par **Real**; ligne 41, lire : **PROGRAM Compatibilité(input,output);**
P. 1555, colonne 1, ligne 28, lire : **4.0** (au lieu de « 4,0 »); ligne 47, lire : **PlusPetiteValeurSignificative** (en un mot).
P. 1556, colonne 1, ligne 15 :
octet:0..255; [sous-ensemble d'entiers]; ligne 16 : **{sous-ensemble de caractères}**.
Colonne 2, encadré, remplacer « sorties » par **output**; « première » par **premier**; « suivante » par **suivant**; « réel » par **real**; « entier » par **integer**; ajouter un point-virgule après **END**; mettre les commentaires entre accolades; ligne 53, lire **UNTIL abs(...)<Epsilon**.
P. 1568, colonne 1, listage :
750 IF OA(3)=0 THEN 760
760 IF OA(4)=0 THEN 770
770 IF OA(5)=0 THEN 780
792 ...:MO;" PIECES D'OR"
Colonne 2 : 3115 IF PS<>"O" AND...
P. 1572, colonne 1, ligne 35, ajouter un point-virgule après **END**.
P. 1573, remplacer « ' » par ' dans tous les listages.
Encadré gris, corriger comme suit :
PROGRAM AuFût (Output); remplacer : « réel » par **Real**; « rousses » par **rousse**; « 0,5 » par **0.5**; « 0,4 » par **0.4**; « END; CASE » par **END; [CASE]**; « Write (arrondir) » par **Write ('arrondir');** « 0,5 » par **0.5**.
P. 1578, colonne 1, ligne 25, lire : 3.141592.
P. 1579, colonne 2, ligne 13 :
50 PRINT "N'EST PAS BISSEXTILE"
Ligne 20 : 30 PRINT "N'EST PAS BISSEXTILE"
P. 1588, colonne 1, listage :
4078 ...:"MORT!!!!!"":GOTO 4099
P. 1594, colonne 2, ligne 22, remplacer « vrai » par **true**; ligne 23 : « faux » par **false**.
P. 1596, colonne 1, ligne 18 : remplacer « (entrées, sorties) » par **(input,output)**; ligne 23 : « ENSEMBLE DE » par **SET OF**; ligne 30 : « entier » par **integer**; ligne 31 : « booléen » par **boolean**; ajouter un point-virgule à la fin de la ligne 39. Colonne 2, lignes 20 et 21 : remplacer « numéro » par **nombre**.
P. 1613, encadré jaune, corriger les lignes du listage :
5316 ...SEMAINE";WK
5489 EW = EW + ((800-X)/800)
P. 1614, listage bas de page, corriger :
numéro:1..999;
occupée:boolean
END; [chambre]
P. 1615, encadré vert :
PROGRAM CoupMax (output);
END.
P. 1616, colonne 1, ligne 8, remplacer « nombre » par **numéro**.
Encadréjaune : **PROGRAM Longueurs (input,output);**
P. 1626, colonne 2, ligne 1 :
CompteCar = ARRAY['A'..'Z'] OF Integer;
ligne 13 : **Liste['M'];** ligne 22 : **Liste['deuxième'];** ligne 27 : **lettre:char;** ligne 30 : **FOR lettre = 'A' TO 'Z';** ligne 34 : **compteur['a'];**
P. 1627, colonne 1, ligne 5 : **OF char**; ligne 21 : **PACKED ARRAY [1..N] OF char;**
Colonne 2, ligne 6 : **chaîne = PACKED ARRAY [1..10] OF char;**
P. 1628, colonne 1, ligne 9 : **write('Chaîne (T pour Terminer)');** ligne 11 : **WriteLn('La chaîne était:','S,','');** ligne 19, remplacer « s » par **S**.
Encadré gris, listage, colonne 1, ligne 36 :
PROGRAM ListeCrible (output);
Colonne 2, ligne 20 :
WriteLn('Crible Erathosthène:50); ligne 39 :

compte = succ(compte); ajouter un point-virgule à la fin des lignes 52, 53 et 54.
P. 1654, encadré bleu, listage, corriger les lignes :
875 IF HR(3) = .5 AND RND(1)<.5 THEN...
5685...:"SEMAINES"
5965 SS = "VOUS...
5985 SS = "IL...
6115 IF LEFTS(...
6167 SS = "10 KG DE VIANDE...
Encadré « Variantes de basic » :
875...:RND(1)<.5 THEN...
P. 1658, colonne 2, listage : ligne 11 : **PROGRAM Incomplet(input,output...;** ligne 14 : **Ouvrir (fichierdes-données);** ligne 19 : **END;** ligne 20 : **END;** ligne 35, remplacer « vrai » par **true**.
P. 1659, colonne 1, lignes 7 et 9, remplacer « entier » par **integer**. Encadré gris : **PROGRAM Portée;** remplacer « entier » par **integer**; « réel » par **real**.
P. 1660, encadré vert, remplacer « entrées, sorties » par **input,output**; « entier » par **integer**; « booléen » par **boolean**; « TABLEAU » par **ARRAY**; mettre les commentaires entre accolades ({}).
P. 1664, colonne 1, ligne 16 :
FUNCTION Odd(nombre:integer):boolean;
Ligne 31, remplacer « vrai » par **true**; lignes 32 et 40, remplacer « faux » par **false**.
P. 1665, listage colonne 2, remplacer « données » par **data**.
Encadré bas de page, remplacer « Postnum » par **Postnom**; « entrée, sortie » par **input, output**; « booléen » par **boolean**; « lit » par **read**; après « WriteLn », écrire les messages entre apostrophes : **WriteLn('...');**
P. 1673, colonne 2, ligne 22, supprimer les deux parenthèses fermantes à la fin de la formule qui devient :
((X+Y)*10)+5.
P. 1674, listage, lignes 6580 et 6585, remplacer « 'o' » par **"O"**.
6730 Z = ((X+Y)*10)+5
P. 1677, ajouter des guillemets (") à la fin des lignes 2030, 2060, 2070, 2080, 2090, 2100, 2110 du listage; ligne 5030, remplacer « 'o' » par **"O"**.
P. 1679, colonne 2, ligne 18, ajouter une parenthèse fermante : **SC = ...AND 3))**.

Volume 8

P. 1684, colonne 1, ligne 12, ajouter un point après « END ».
Colonne 2, ligne 29, remplacer « entrée, sortie » par **input, output**.
P. 1685, colonne 2, ligne 37, remplacer « texte » par **text**; ligne 41 : « booléen » par **boolean**.
P. 1686, colonne 1, lignes 27 et 28, remplacer « entrée » par **input**; « sortie » par **output**.
Colonne 2, listage, remplacer « texte » par **text**; mettre les commentaires du programme entre accolades.
P. 1694, listage, compléter la ligne :
6935 ...:THEN X = 1:T = 16
P. 1696, ajouter des guillemets à la fin des lignes 130, 140, 150 et 160 du listage.
P. 1714, colonne 1, ligne 19, lire :
TableauLong = ARRAY 1..100 OF real; ligne 30, lire : **nombre:integer;**
Colonne 2, ligne 16 :
PROGRAM DeuxPlusDeux(output);
Lignes 18 et 22, remplacer « entier » par **integer**.
P. 1715, colonne 1, ligne 47 : **car:char**;
Colonne 2, ligne 11 : **Ligne = ligne -suivant**
P. 1716, listage, remplacer « booléen » par **boolean**; « texte » par **text**; « entrée » par **input**; « sortie » par **output**.
P. 1725, colonne 1, ligne 11, remplacer « TS() » par **TSO**.
P. 1726, colonne 1, ligne 6, remplacer « (chars) » par **(char)**.
Dans le listage, remplacer « entrées » par **input**; « sorties » par **output**; « booléen » par **boolean**; « faux » par **false**; « écrire » par **write**.
P. 1727, même chose; mettre les commentaires entre accolades.
P. 1728, listage : remplacer « variant » par **variante**; « booléen » par **boolean**; « faux » par **false**; « vrai » par **true**; « vrai, mâle » par **true, masculin**; « CASE sexe:sexe OF chaîne) » par **CASE sexe OF chaîne**.
P. 1747, colonne 2, encadré, remplacer le numéro de ligne (à la 10^e ligne) par **10078** (au lieu de « 10070 »); à la fin de la ligne 10145, écrire un point-virgule (au lieu du deux-points).
P. 1763, corriger les lignes du listage :
140 IF H1<1 OR H1>2 THEN 130
600 IF D(D+1)>A(D) THEN...
P. 1830, colonne 3, à la fin des lignes 1015 et 1020 du listage, lire : **TAILLE%** (sans espace).
P. 1834, colonne 1, ligne 8 : **...:IF IS="" THEN...**
P. 1843, colonne 1, ligne 54, remplacer « to » par **10** (2 et 10).
P. 1844, encadré « Variantes de basic », à la fin de la ligne 4090, lire : **...3300:LET Y = SK**.
P. 1857, colonne 1, ligne 13, lire : **(SETQ B(TIMES...)**
P. 1883, corriger la ligne 360 du listage :
360 ...AND 2-(C-1))>0 THEN...